



**University of
Zurich^{UZH}**

**Zurich Open Repository and
Archive**

University of Zurich
University Library
Strickhofstrasse 39
CH-8057 Zurich
www.zora.uzh.ch

Year: 2013

Airborne prism experiment calibration information system

Hueni, Andreas ; Lenhard, Karim ; Baumgartner, Andreas ; Schaepman, Michael E

Abstract: The calibration of remote sensing instruments is a crucial step in the generation of products tied to international reference standards. Calibrating imaging spectrometers is particularly demanding due to the high number of spatio-spectral pixels and, consequently, the large amount of data acquired during calibration sequences. Storage of these data and associated meta-data in an organized manner, as well as the provision of efficient tools for the data analysis and fast and repeatable calibration coefficient generation with provenance information, is key to the provision of traceable measurements. The airborne prism experiment (APEX) calibration information system is a multilayered information technology solution comprising a database based on the entity-attribute-value (EAV) paradigm and software written in Java and MATLAB, providing data access, visualization and processing, and handling the data volumes over the expected lifetime of the system. Although developed in the context of APEX, the system is rather generic and may be adapted to other pushbroom-based imagers with little effort.

DOI: <https://doi.org/10.1109/TGRS.2013.2246575>

Posted at the Zurich Open Repository and Archive, University of Zurich

ZORA URL: <https://doi.org/10.5167/uzh-77303>

Journal Article

Published Version

Originally published at:

Hueni, Andreas; Lenhard, Karim; Baumgartner, Andreas; Schaepman, Michael E (2013). Airborne prism experiment calibration information system. *IEEE Transactions on Geoscience and Remote Sensing*, 51(11):5169-5180.

DOI: <https://doi.org/10.1109/TGRS.2013.2246575>

Airborne Prism Experiment Calibration Information System

Andreas Hueni, *Member, IEEE*, Karim Lenhard, Andreas Baumgartner, and Michael E. Schaepman, *Senior Member, IEEE*

Abstract—The calibration of remote sensing instruments is a crucial step in the generation of products tied to international reference standards. Calibrating imaging spectrometers is particularly demanding due to the high number of spatio-spectral pixels and, consequently, the large amount of data acquired during calibration sequences. Storage of these data and associated meta-data in an organized manner, as well as the provision of efficient tools for the data analysis and fast and repeatable calibration coefficient generation with provenance information, is key to the provision of traceable measurements. The airborne prism experiment (APEX) calibration information system is a multilayered information technology solution comprising a database based on the entity–attribute–value (EAV) paradigm and software written in Java and MATLAB, providing data access, visualization and processing, and handling the data volumes over the expected lifetime of the system. Although developed in the context of APEX, the system is rather generic and may be adapted to other pushbroom-based imagers with little effort.

Index Terms—Imaging spectroscopy, relational database, sensor calibration.

I. INTRODUCTION

REMOTE sensing technologies have the potential of acquiring data with a spatial coverage, temporal resolution, and continuity, which allow the parameterization of earth system science models at regional and global scales. Such remotely sensed data are referred to as fundamental climate data records (FCDRs). These basic data are subsequently transformed into end-user products describing essential climate variables (ECVs) by data assimilation [1]. Of the 44 ECVs identified in the Global Climate Observing System Second Adequacy Report [2], a total of 25 are largely dependent on satellite observations, effectively rendering remote sensing instruments one of the most important means of data collection for earth system sciences.

Of the multitude of available sensor systems, the family of imaging spectrometers exhibits a high potential for the retrieval of ECVs from all spheres of the climate system [3], [4]. While some spaceborne imaging spectrometers do exist [5], [6] or are planned, [4], [7]–[9], most instru-

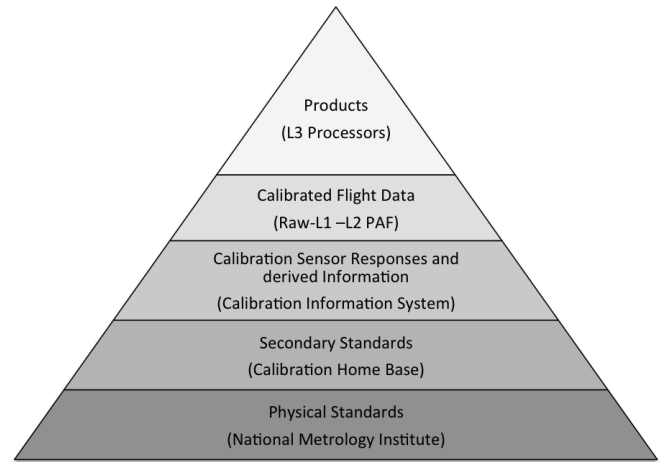


Fig. 1. DIKW applied to remote sensing system calibration and product generation.

ments [10]–[13] are currently deployed on airborne platforms [3].

The calibration of imaging spectroradiometers and their data is technically demanding due to the high number of spatial/spectral pixels, and also hampered by the notion that spectroradiometer measurements are still considered as one of the least reliable of all physical measurements [14], [15]. Calibration is an essential and critical step before higher product generation can be achieved with the accuracies required for the successful parameterization of climate models in order to reduce the uncertainties of predictions [16].

Imaging spectrometers generally measure at-sensor radiances by reference to International System of Units (SI) through either laboratory or vicarious calibration. System calibration allows defining the traceability of measurements to the SI standard and hence enables the comparison of data stemming from different sensor systems. Such traceable calibration forms the basis for the generation of consistent geophysical and biophysical products [17].

These dependencies may be conceptualized using an adapted data–information–knowledge–wisdom (DIKW) [18], [19] representation (Fig. 1). DIKW is a model describing the building of knowledge from information based on facts or data and hence has found use in multidisciplinary research, ranging from philosophy to systems analysis [20], [21]. Various flavours of the DIKW do exist [22], and for the purpose of this research, we add the notion of signals [23] while omitting the rather elusive tier of wisdom [24].

Manuscript received September 29, 2012; revised December 2, 2012; accepted January 22, 2013.

A. Hueni and M. E. Schaepman are with the Remote Sensing Laboratories, University of Zurich, Zurich 8057, Switzerland (e-mail: ahueni@geo.uzh.ch; schaep@geo.uzh.ch).

K. Lenhard and A. Baumgartner are with the Remote Sensing Technology Institute, Oberpfaffenhofen 15236, Germany (e-mail: karim.lenhard@dlr.de; andreas.baumgartner@dlr.de).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TGRS.2013.2246575

The lowest tier is formed by the physical standards, i.e., SI units, as provided by national metrology institutes. Here, instrumentation used in calibration laboratories may be calibrated and thus made traceable; these are the secondary standards, equivalent to tier 2. This second tier produces optical stimuli leading to calibration sensor responses and information derived thereof. These data and information constitute the third tier and are held by a component we refer to as the calibration information system (CAL IS). Calibrated flight data (tier 4) are based on calibration information and are generated by processing and archiving facilities (PAFs) [25]. Products are the output of higher level algorithms and form the top of the pyramid, being equivalent to knowledge as it adds actionability to information [22].

We define a CAL IS as a system layer that stores raw sensor calibration and characterization data and generates information describing the instrument's electrooptical chain that converts signals from a continuous electromagnetic space into digital numbers within a discrete space. The CAL IS produces calibration coefficients used by the PAF to calibrate flight data and by this establishes the traceability link between airborne data and physical standards. The CAL IS holds information that leads to an enhanced understanding of the sensor properties and characteristics and as such supports the calibration scientists in developing their system knowledge.

This paper defines the data sources generating raw calibration and characterization data, lists the requirements for a CAL IS, and documents the chosen implementation. The system is targeted at the airborne prism experiment (APEX) system, but the general concept essentially applies to any frame based imaging system.

II. DATA SOURCES

A. APEX

The European Agency's (ESA) airborne imaging spectrometer APEX was developed under the PROgramme de Développement d'EXpériences scientifiques (PRODEX) program by a Swiss–Belgian consortium with the concept phase starting in 1998 and leading to a first test flight in 2008. APEX was formally accepted by ESA at the end of 2010 and entered the exploitation phase in 2011. It features up to 532 spectral bands in full spectral mode, ranging from 375 to 2500 nm. Spectral programmability of the VNIR sensor enables achieving higher signal-to-noise ratios (SNRs) by reducing the number of bands in a binned configuration. Data are acquired in 1000 pixels across track with a field of view (FOV) of 28°, resulting in ground pixel sizes of 1.5–2.5 m at typical flight levels of 3000–5000 m above ground.

The main components of the APEX system are: 1) an optical subunit (OSU) containing the optoelectronics; 2) a control and storage unit (CSU) comprising the instrument control computer, solid state devices for the storage of the data stream, and a positioning system; and 3) a temperature control unit (TCU) responsible for the regulation of the OSU optical base plate temperature to a stabilized 20 °C.

Average Data Volumes per Calibration Experiment [GB]

■ Std. RAD ■ Std. GEO ■ Std. SPEC ■ Special Exp

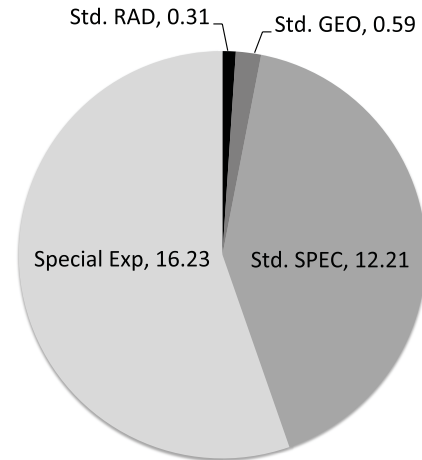


Fig. 2. Average data volumes acquired during CHB missions per calibration experiment [GB].

TABLE I
NUMBER OF META-PARAMETERS PER FRAME FOR THE
MOST COMMON CALIBRATION EXPERIMENTS

Experiment	Meta-Parameters Per Frame
Absolute radiometric calibration	157
Relative radiometric calibration (flat fielding)	158
Spectral calibration	217
Across-track geometric calibration	204
Along-track geometric calibration	204

Image data are stored frame-wise, where a frame consists of a combined readout of VNIR and SWIR detectors. The storage size per frame depends on the chosen binning pattern. Frames in the default binning mode amount to storage sizes of 0.62 MB and to 1.1 MB in the unbinned mode. Meta-data per frame contain information on instrument settings (e.g., integration time) and readings of various auxiliary sensors (pressure, temperature, and voltages) mounted in the OSU. A total of 88 meta-parameters are recorded for each frame by the CSU.

B. APEX Calibration

APEX calibration refers to the calibration and characterization of APEX in the calibration home base (CHB) with the goal of collecting data allowing the radiometric, geometric, and spectral calibration of the instrument [11]. These data essentially form the base for the estimation of calibration coefficients that are applied to imaging data during data calibration. Standard system calibration runs generate around 13 GB of data, while special experiments on average double the standard data volume (Fig. 2). CHB calibration missions take place once to twice a year, resulting in a total raw data size of around 300 GB accumulated since 2007. Data are stored on the file system of the CSU in automatically generated hierarchical folder structures with a naming convention for both folders and files.

Calibration settings are sent to the CHB equipment and these parameters are also added to the meta-data file associated with each frame. The total number of meta-parameters per frame generated during calibration depends on the calibration experiment, comprising APEX system parameters and CHB settings as provided by the CHB interface (Table I).

C. Calibration Home Base

The CHB was commissioned by ESA for the calibration of APEX [26]. However, the laboratory can also be used for other sensors since imaging spectrometers used in optical remote sensing often display similar properties. The CHB provides defined light sources in the typical wavelength range of these instruments: integrating spheres for the radiometric calibration, and a monochromator assembly and a slit-collimator assembly in combination with a rotating mirror on a translation stage to generate optical stimuli for the spectral and geometric calibration, respectively.

Since the spectral and geometric calibration procedures can consist of several hundreds or thousands of short measurements with different settings of the involved light source, the CHB was set up with automation kept in mind. The software controlling the calibration procedures was designed based on the master/slave pattern. The slave controls the CHBs devices and light sources, while the master runs calibration procedures by requesting CHB settings from the slave and managing the data acquisition of APEX. Generally speaking, the master provides an interface to define measurement sequences, with specific routines being written for each specific calibration experiment.

The communication between master and slave follows a synchronous client/server model, in which the slave takes the role of the server and receives requests by the master [27]. Each request for CHB settings generates exactly one reply, which indicates that the CHB has assumed the requested state and is ready for measurements. This reply contains all the meta-data generated by the CHB. These meta-data stem from the laboratory devices, e.g., specifying the wavelength to which the monochromator is set, and from an environmental sensor that supplies room temperature, atmospheric pressure, and humidity.

The communication between master and slave takes place over a TCP/IP connection, allowing operating master and slave on separate computers. The data are exchanged in the form of eXtensible Markup Language (XML) files, which are human-readable and well suited for the small amount of data exchanged, which is on the order of 1 kB per request/reply. An additional advantage of using XML files is that their content can be checked for consistency against an XML schema definition (XSD) file. This includes checks for the completeness of the parameters, their data type, and their valid ranges.

III. REQUIREMENTS

APEX CAL IS requirements are mainly based on the goals of generating system calibration information and enhancing the knowledge about the system in general, as well as on the expected data volumes over the nominal lifetime of the system.

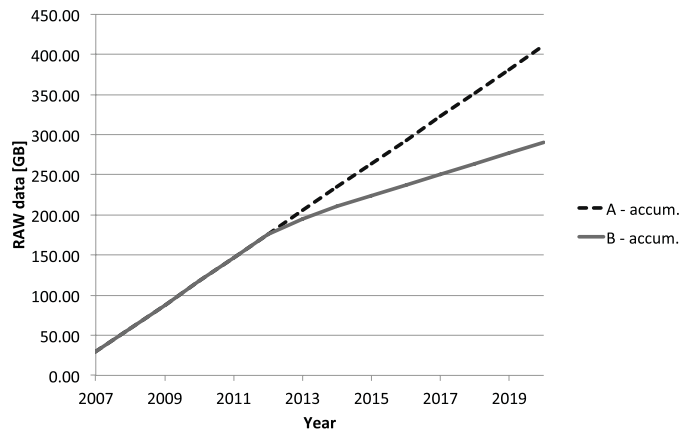


Fig. 3. Estimated RAW data volumes for two scenarios over the expected system lifetime.

A. Data Volumes

The CAL IS must be able to handle the estimated total data volume over the expected lifetime of the system. The data volume includes calibration data acquired during the operational stage, currently set at 10 years, plus data acquired during the system acceptance test phase. Raw data volumes range between 290 and 410 GB under two different scenarios (Fig. 3): 1) per-annum volumes remain identical to the current average, i.e., special experiments constitute half of the volume and 2) special experiment volumes diminish exponentially over time, leading to per-annum volumes mainly governed by the standard data calibration runs.

These estimated volumes will roughly double when data are processed to level-1 in the CAL IS, i.e., leading to total sizes in the order of 0.6–0.8 TB.

B. Generic Frame Support

The number of spectral pixels of an APEX frame depends on the binning patterns applied to the VNIR detector. The CAL IS must be able to seamlessly handle frames of differing binning patterns, including the frame-size-independent storage and the generation of calibration coefficients for various binning patterns.

C. Flexible Meta-Parameters

A flexible handling of the number of meta-parameters per frames is required because: 1) the number of meta-parameters is dependent on the particular calibration experiment; 2) the CAL IS can add new meta-parameters generated from both meta-data and frame data during the forming of information; and 3) upgrades in the CHB may lead to different or additional meta-parameters over time.

D. Data Ingestion, Data Structuring, and Quality Control

Data ingestion must be an automated process, retrieving frame and meta-data from the files generated during calibration. Near-real-time data control during calibration missions requires loading data into the CAL IS at various points in time, i.e., allowing data ingestion with a delta data loading capability.

The data hierarchy generated during the calibration campaign on the CSU reflects the experimental structure and provides an easy way for the users to interactively navigate through the wealth of calibration data. The CAL IS must retain such structures and replicate them within the database.

Automated generation of quality flags is important for two reasons: 1) detection of problems in near-real time during calibration runs and 2) exclusion of unsuitable data from calibration coefficient generation processes. Examples include saturation detection or thresholds for system temperatures and pressures.

E. Support for Processing Levels

The concept of processing levels is identical to the one commonly implemented in airborne and space-based instrument PAFs. The levels reflect various stages of processing in the system and allow the efficient generation of higher level products without a complete reprocessing starting from raw sensor data. Storage of several processing levels allows the easy study of effects caused by transforming processes, helping the debugging of according algorithms. Table II lists the required processing levels of the APEX CAL IS.

F. Interactive Data Exploration

Developing a sound knowledge about the sensor system and controlling the generation of calibration coefficients require the ability to graphically explore data in an interactive fashion. The dimensions to be explored are: 1) the spatio-spectral pixel values of a frame 2) the time domain as the system response changes due to modification of or noises in the external stimuli or due to system-inherent drifts or noises 3) the meta-data space [28] and 4) combined meta-data–spatio-spectral domains where the spectral response at any pixel may be mapped versus parameters of the meta-data space.

G. Provenance

Provenance describes the origin and evolution of data [29]. This information is important for the APEX CAL IS, as it allows tracing effects found at any processing stage to its original cause. Provenance data forms a graph consisting of data sources, data sinks, and processes. Such a topology is also highly useful when the definition of uncertainty budgets is required, as all the contributing sources of noise are essentially given by the provenance graph.

An example is the level-0 to level-1 processing, i.e., the dark current correction: each level-1 frame must have an associated creating process description and links to both level-0 frame and dark current frame used for the correction.

IV. CONCEPTS

A. Overall Architecture

The APEX CAL IS is designed as a multilayered system (Fig. 4). A relational database management system (RDBMS) serves as storage solution, implemented by a MySQL server (Version 5.5). Data are stored in physical database tables within the RDBMS, using a mixture of traditional relational

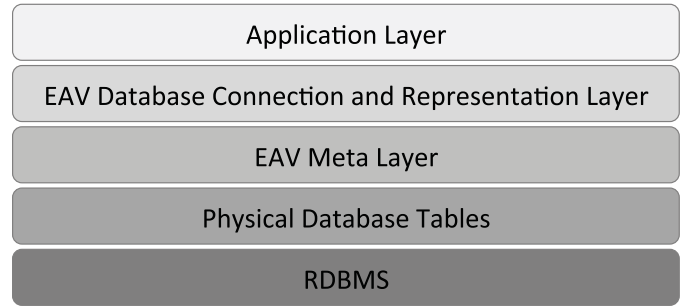


Fig. 4. APEX CAL IS layers.

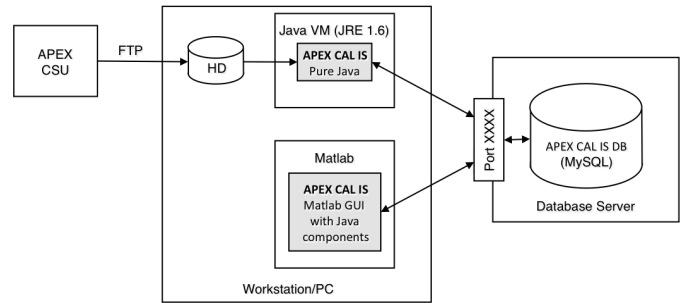


Fig. 5. APEX CAL IS system architecture and dataflow.

database model and of a meta-system architecture known as the EAV paradigm [30]. The EAV meta-layer is the representation of the meta-data known to the APEX CAL IS by according entries in physical database tables. The database connection and representation layer is written in Java and handles all communications with the database, offering functionality for data insert, querying, and deletion, essentially mapping the EAV information to an object-oriented representation and representing frame data as matrices. The application layer holds routines for the analysis and processing of data, including the graphical representation. This layer is implemented in MATLAB (Release 2010a) using Java components for the communication with the database and for some graphical data representations.

Fig. 5 illustrates the dataflow and the overall system architecture. Frame and meta-data files are transferred from the APEX CSU to a workstation by FTP. A pure Java-based application is used to ingest these files into the APEX CAL IS database, which is hosted by a database server. Higher level processing, visualization, and analysis are carried out in a MATLAB environment, relying on Java components for database communication, i.e., on lower level data services as implemented in the EAV database connection and representation layer. The illustrated setup reflects the most common one, but installations where a laptop takes the role of a database server and processing computer at the same time are also feasible, e.g., within the CHB where a direct feed into a remote database server may not be as performant as a locally hosted database instance. The centrally hosted database allows the simultaneous data access by several researchers.

TABLE II
APEX CAL IS PROCESSING LEVELS

Designator	Name	Description
Level-0	RAW	Raw frame data as generated by APEX
Level-1	DC Corrected	Frame data corrected for the dark current, taken from the closest dark-current frame in the acquisition time line
Level-2	Desmeared	Frame data corrected for image smear caused by the readout mechanism (applies only to the VNIR part of the frame)
Level-3	Intermediate system coefficients	Coefficients describing the electrooptical chain, but not yet integrated for all spatio-spectral pixels, e.g., gains/offsets for the nadir pixels
Level-4	System calibration layers	Coefficients for each spatio-spectral pixel, i.e., a matrix of the same dimension as the imaging frame and directly applicable in the data calibration process within the APEX PAF

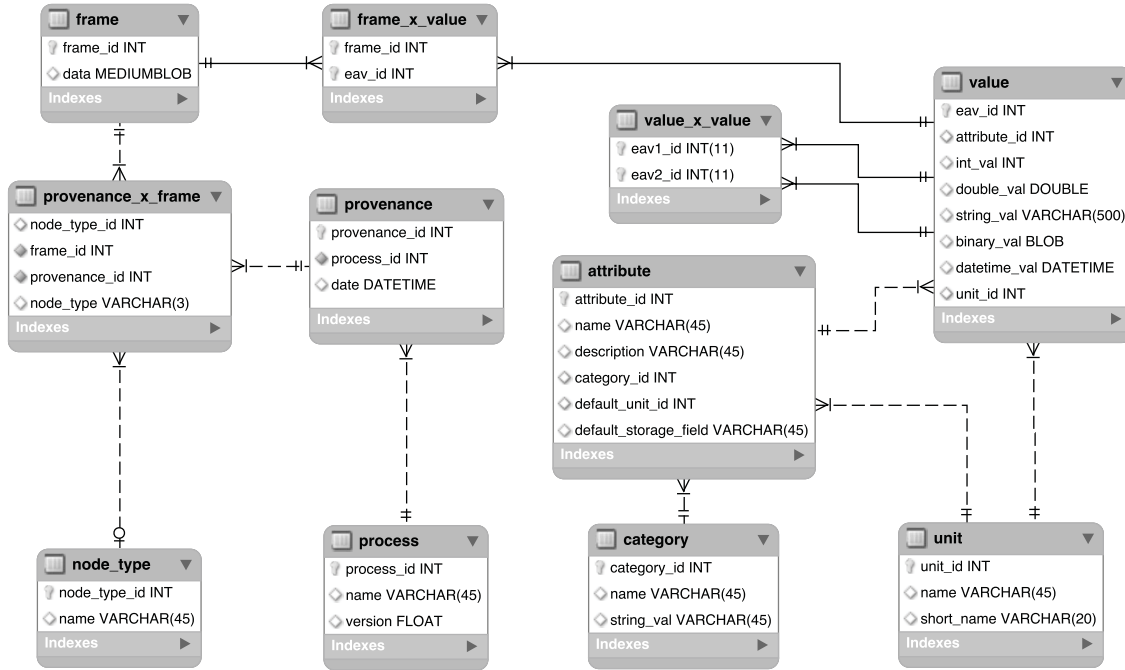


Fig. 6. Entity relationship diagram of the APEX CAL IS schema.

B. Database Schema

The APEX CAL IS schema (Fig. 6) implements the EAV paradigm, but uses some traditional relational modeling as well, as suggested in [31]. The frame table represents the primary data, i.e., the entities. Most of the entries in this table are APEX frames, but as data are stored as binary large objects, data of other dimensionality may be stored as well. Frame data are serialized objects of matrix classes belonging to the UJMP package [32], and as such may assume dimensionalities between 1 and 3, referring to single spectra, 2-D frames, and imaging cubes, respectively.

Frames can be associated with multiple meta-parameters, which in turn may be referenced by multiple frames. This is achieved by a cross-relational table (*frame_x_value*) and a value table holding the actual values. A tuple within the value table may assume the data types of integer, floating point, string, binary object, or date/time, storing them in the applicable fields, i.e., adopting one possible representation of the values within an EAV schema [31]. Value tuples can refer to other value tuples by the way of the *value_x_value* cross-relational table. This is heavily used in the modeling

of hierarchical folder structures in the system, while the representation of these relations is part of the system software.

Value tuples refer to both attribute and unit table entries. The APEX CAL IS handles attributes and units in a flexible way, allowing for values of a certain attribute to have differing units. To support the use of the EAV-related Java classes in other projects where a more strict approach is needed, namely the SPECCHIO spectral database project [33], the option to define standard units and default storage fields was added to the attribute table.

Provenance is modeled by a provenance table, representing instances of transformations. A transformation comprises a processing module of a certain version, stored in the process table, and a number of input and output frames, cross-linked via the *provenance_x_frame* table and the input/output node type given by entries in the *node_type* table.

Data integrity is ensured by foreign keys and the corresponding constraints in the database schema.

C. Data Insert

Data insert deals with the ingestion of data stored on the file system and with the insert of processing output.

Meta-data inserts are carried out as bulk inserts, i.e., all rows that are inserted for a frame into the values table and frame_x_values table, respectively, are combined into one SQL statement, thus minimizing the database statement overhead and leading to an optimized loading speed.

Frames generated by processing existing frames in the database, e.g., dark current correction, are inserted into the database by adding a new row in the frames table and linking with existing meta-data of the input frames, thus avoiding data redundancy. Meta-parameters not applying to the higher level frames are removed or updated, thus either omitting a link between new frames and original meta-parameters, or adding new meta-parameters with updated values.

Generally speaking, all data retrieval is based on meta-data subspace projections, i.e., frames are selected by defining meta-parameter restrictions and the frames complying with these are contained within a subspace [28]. In practice, two variations for the definition of such projections exist: 1) selection via browsing of the data hierarchy where frames are identified by either their filenames or their containing directories and 2) the programmatic definition of SQL queries that convert the restrictions to actual statements. For many instances of data processing or analysis, data selection involves both methods by first selecting a set of frames in the interactive browser and then projecting that set to a subspace by additional restrictions. Fig. 7 illustrates such a combined use on the example of the radiometric calibration in the form of a sequence diagram.

```

sequenceDiagram
    participant App as Application Layer
    participant EAV as EAV DB Con. & Rep. Layer
    participant Radiometric as Radiometric Cal (Matlab)
    participant Query as Query Builder
    participant Spectral as Spectral Data Browser
    participant AV as AV_Matching ListCollection
    participant APEX as APEXCallIS Services
    participant Frame as Frame
    participant Data as Data Processor

    App->>Radiometric: Update_tree_based_query
    App->>Query: Add_condition(level 1)
    App->>Query: Get_frame_ids
    App->>Spectral: insert_into_lists(ids, N=Filter)
    App->>AV: Get_ids(group_no)
    App->>APEX: Load_frames(group_id)
    App->>Frame: Frame(rad_cal_layers)
    App->>Frame: Set_metadata(metadata)
    App->>Frame: insert_frame_and_metadata()
    App->>Data: add_provenance_record(time, input_ids, output_ids)
    App->>Data: insert_provenance_data()
  
```

group data by multiple attributes. The Table III lists some of the main methods/classes and examples of their practical application.

Data processing relates to the transformation of data, either for analysis purposes or for the generation of calibration products. Data volumes are rather big, and efficient procedures are required to select and load data from the database and insert possible results. In this respect, the number of database statements must be minimized while ensuring that the memory allocation is sufficient to hold the data to be processed.

A further strategy is the partitioning of larger datasets during processing by loading only subsets into memory. This division into data collections is, e.g., applied when processing spectral calibration data, where a full calibration dataset may be several gigabytes in size.

F. Data Representation

The graphical representation to the user is key to the efficient handling of these multidimensional data. Developing a graphical user interface (GUI) in MATLAB while using

TABLE III
MAIN METHODS/CLASSES FOR DATA RETRIEVAL, LOADING, AND GROUPING

Method/Class	Use/Example
<code>get_list_of_metaparameter_vals</code> (frames, attribute)	Return a list of values for a certain attribute for a set of frames, e.g., list of integration times per frame
<code>get_eav_ids</code> (frame, attributes)	Return all <code>eav_ids</code> of a frame, optionally restricted to an attribute set. Used to retrieve a complete metadata set or a metadata subset of a frame
<code>get_eav_ids</code> (frames, attribute)	Generic function to return all <code>eav_ids</code> of a set of frames for an attribute, optionally distinct: e.g., <code>get_eav_ids</code> holding the neutral density filter value used in absolute calibration
<code>get_closest_product_frames</code> (frame, product)	Returns frames that are products (level-3 and level-4) and are closest on a time line to the supplied frame: e.g., used to identify spectral calibrations to apply in convolution operations required for radiometric calibration
<code>get_frames_by_attribute</code> (frames, attribute, value)	Returns the frames of a set of frames that have an attribute with a value matching the supplied one: e.g., select frames out of a subset that have a certain neutral density filter value
<code>frame_bulk_reading</code> (frames)	Read frames into memory using one database statement for speed reasons. Used to build spectral cubes on the fly
<code>AV_MatchingListCollection</code>	Class to group a set of frames by multiple attribute values: e.g., used to group dark current frames by integration time to build a dark current system model or grouping of absolute calibration frames by neutral density filter as required for radiometric calibration

Java Swing components allows for the seamless integration of the functionality offered by the EAV database connection and representation layer with the advanced plotting tools of MATLAB. A Java key component is the hierarchical data browser that graphically represents the structure of the data as stored in the database as recursive attribute–value entries. The integration of Swing components in the MATLAB GUI is accomplished by the JControl package [34].

Frame data within the MATLAB environment are represented as matrices. These are populated by first loading the frames from the database into Java where they are deserialized and exist as UJMP instances. In a second step, data are transferred into MATLAB matrices by using a UJMP to standard Java double-array conversion.

V. RESULTS

This section demonstrates the capabilities of the system by documenting the loading speed for data retrievals and the graphical data representation on the example of the main application user interface. A detailed description of the individual calibration and analysis modules is beyond the scope of this paper and will be treated in dedicated future publications. Readers interested in a practical usage example of the APEX CAL IS are referred to the case study presented in Section VI.

Data loading speed was tested by loading frame collections selected in the spectral data browser and ranging between 1 and 600 items into MATLAB, using three different setups to give indicative speeds for the most likely configurations: 1) database server and application running on the same machine, i.e., localhost; 2) database server hosted by a machine in the same Ethernet network as the workstation running the application; and 3) database hosted by a server at our APEX partner institution VITO and the application running on a workstation at the University of Zurich, with a network connection established using VPN tunneling. The

latter setup being the one used for shared database access for both the operations and science teams of the APEX project. At the time of the testing, calibration data for the years 2010–2012 was loaded into the system, consisting of 190 000 frames and 10.2 million meta-parameter entries.

Fig. 8 shows the resulting loading speeds for the three setups: the total loading time refers to the time needed to load frames from the database into MATLAB (top left); the Java loading time is equivalent to the time spent in the EAV database connection and representation layer to load the frame data from the database into the memory of the workstation (top right); the total loading time per frame is the amount of time required to load one frame into MATLAB under the scenario of different collection sizes (bottom left); and the Java loading time per frame is the time spent within Java per frame for the different collection sizes (bottom right).

The results show clearly that the loading speed is a function of the number frames and largely governed by the time spent in the EAV database connection and representation layer, loading the frame data from the database into Java allocated memory. The loading times are also governed by the type of database connection, with the localhost being fastest as it uses a Unix socket file, while the most overhead and delays occur for the tunneled connection to a server in a different physical network.

The loading times per frame are a function of both the number of frames and the database server hosting location. Data loading involves a certain overhead such as the sending of statements and the compilation of meta-data on the frame sizes. Hence smaller frame collections show a notable overhead per frame, which gets minimized as collection sizes increase.

Actual speeds of the system can vary, as they are influenced by the database server configurations such as the memory allocated for the caching of query results, the overall network traffic, the number of other processes running on the application workstation, the amount of RAM available, and

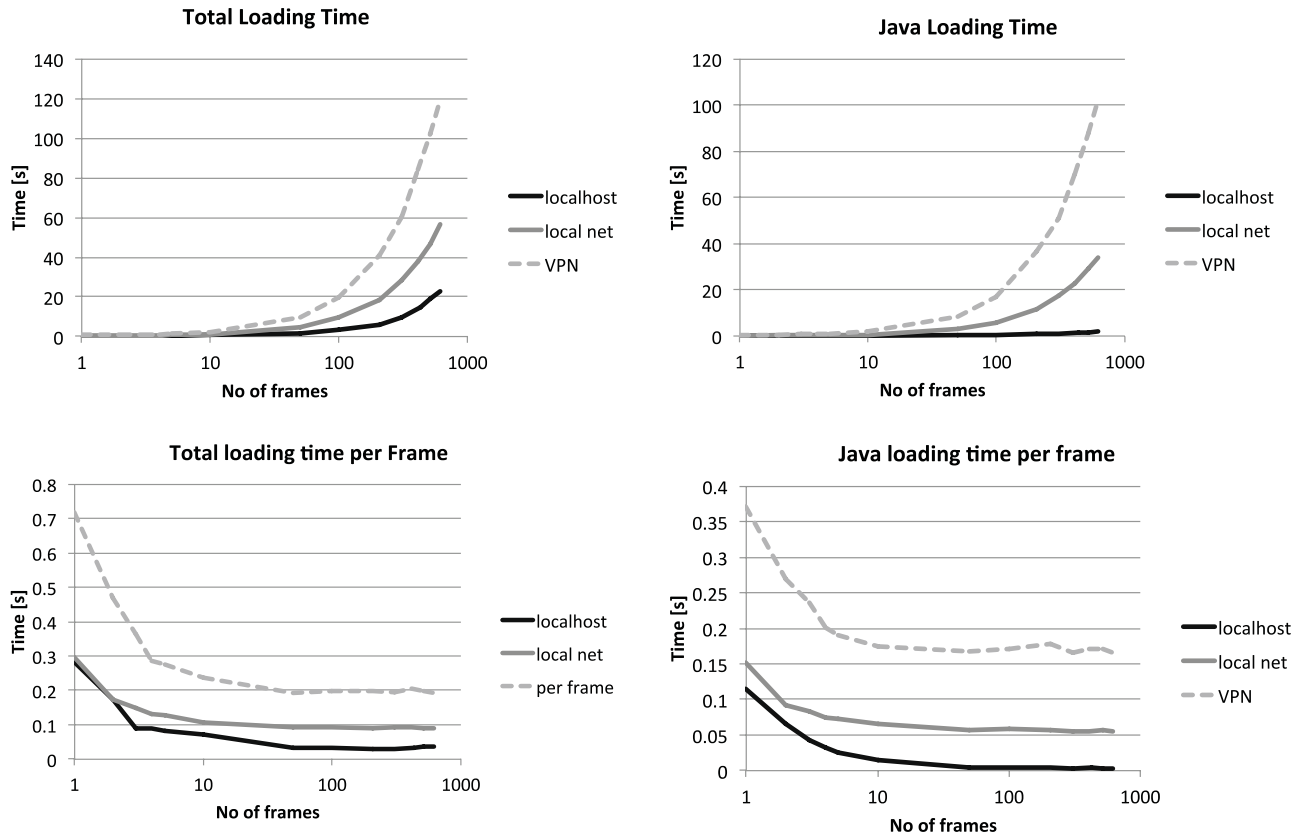


Fig. 8. Total and per-frame loading times from the database into Java and MATLAB.

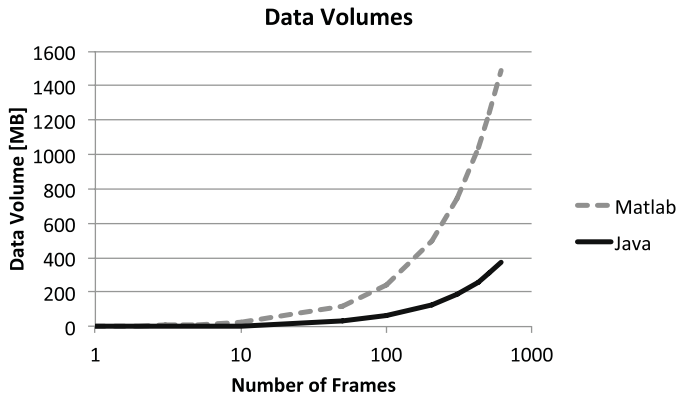


Fig. 9. Frame data volumes when loading frames into Java and MATLAB.

the fragmentation of the current free memory, which impacts the speed of memory allocation within MATLAB. Hence, the shown loading speeds hold true only when enough RAM to hold the whole virtual cube is available, and the performance will drop significantly when the operating system is forced to use virtual memory.

The data sizes involved in the loading speed test are shown in Fig. 9, illustrating the increase in data volume as frames represented by 16-bit integer matrices in Java are moved into MATLAB matrices of 64-bit floating point data type. Frame collection sizes bigger than 150 frames are very rare under usual system usage, with the most common number of frames being loaded ranging between 1 and 50, i.e., the loading speed remains in an agreeable range for the users.

Fig. 10 shows the main interface to APEX CAL IS written in MATLAB. The spectral data browser Java component is featured on the left of the window, showing the data held by the database organized by campaigns, and the directory structure below each campaign reflecting the original storage on disk. Selected data are visualized in the four displays to the right of the data browser, showing a frame view with 1000 pixels across track and spectral pixels depending on the binning pattern (top left), a spectral profile of the selected spatial position (top right), an across-track profile for the selected spectral position (bottom left), and an along-track profile for the selected across-track and spectral position (bottom right). Positions within this virtual cube are selected with three Java-based scrollbars placed at the edges of the frame display and along-track display, respectively. Java radio controls (bottom left) allow focusing the display on VNIR, SWIR, or both detectors. Two smaller data displays (middle right) show the number of saturated pixels per frame of the virtual cube split into SWIR and VNIR detectors, based on saturation data compiled during initial data loading. All further functions of the APEX CAL IS, such as calibration and analysis functions, are accessed via menu entries provided by the main window.

The virtual cube shown in Fig. 10 comprises 150 level-1 radiometric calibration frames acquired with five different radiance settings on a small integrating sphere illuminating the center of the FOV. The intensity steps can be easily discerned in the along-track profile (bottom right plot). The two last intensities exhibit saturation in both detectors as a result of

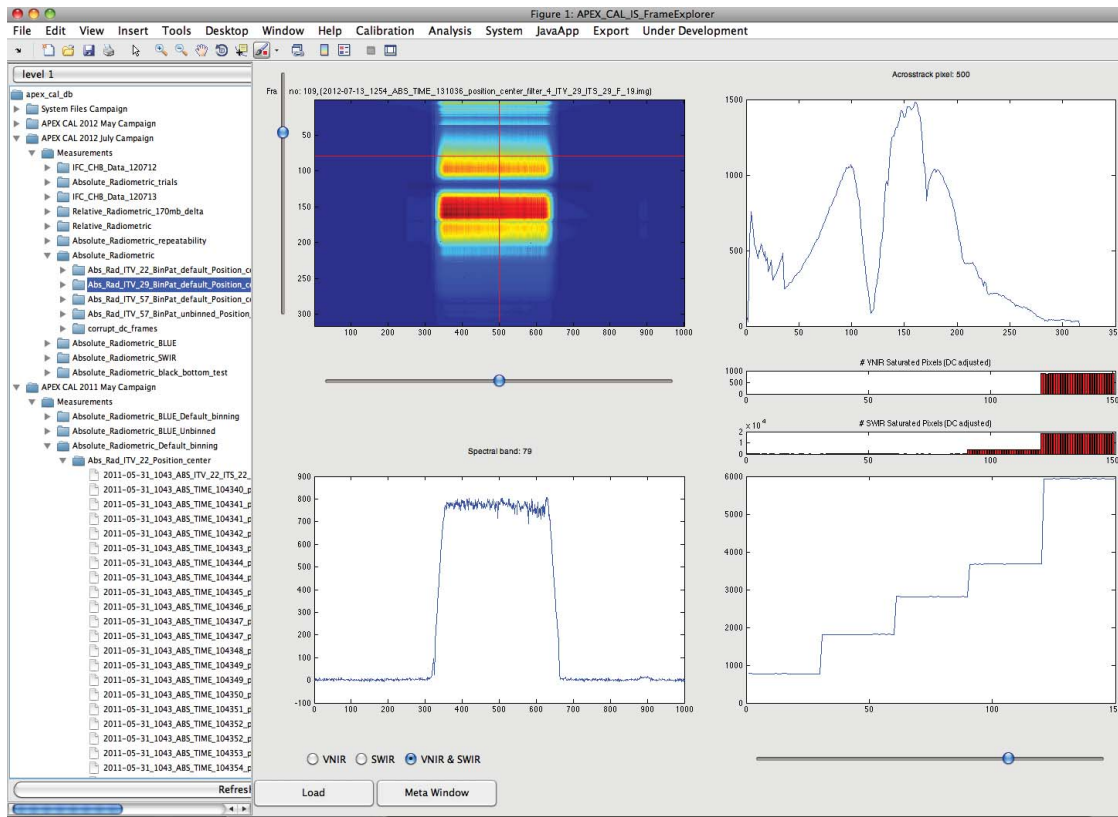


Fig. 10. MATLAB APEX CAL IS main interface.

too high an integration time for these radiance levels. These saturations are indicated in the GUI by the red bars in the two smaller displays (middle right).

VI. CASE STUDY

This section exemplifies the practical application of the APEX CAL IS by alluding to the spectral system calibration, which is rather data-intensive as well as particularly interesting from an algorithmic point of view. Goal of the spectral calibration is the definition of center wavelength and full-width at half-maximum (FWHM) parameters for all spatio-spectral pixels of the system's detectors. The spectral calibration module must primarily deal with three problems: 1) processing of level-1 frames, amounting to 5.8 and 5.3 GB of standard spectral calibration data for VNIR and SWIR sensors, respectively; 2) extracting spectral response functions (SRFs) from these data at the sampled points and providing values for all pixels by suitable inter/extrapolation algorithms; and 3) enabling the user to interactively inspect each stage of the calibration layer generation. The spectral calibration is implemented generically and operates automated and independent of the selected spatio-spectral sampling pattern, spectral binning pattern, and chosen monochromator sampling step size by relying on the detailed meta-data, minutely describing the acquisition and calibration instrument settings.

The overall processing time is largely governed by the loading of the data into the main memory of the client machine, and the generation of spectral calibration layers requires around a quarter of an hour per detector, which is acceptable as operationally performed only once or twice a

year. Data loading is partitioned based on the sampling pattern and thus only a subset of the calibration data is loaded into memory at a time. Once extracted, the actually required data vectors and associated meta-data may be saved in intermediate files, typically taking a mere 300–500 kB of memory per detector.

Fig. 11 presents the GUI of the spectral calibration module. The displays show the following interactive information, essentially representing the extracted data for the VNIR detector (from left to right and top to bottom): (a) DN response of a selected illuminated pixel plotted versus the changing monochromator wavelength; (b) Gaussian curve fitted to the data points, used for the determination of center wavelength and FWHM; (c) spatio-spectral sampling pattern with currently visualized sampling point indicated by red crosshairs; (d) center wavelength across-track profiles, i.e., equivalent to the spectral misregistration, for the measured spatial pixels at the currently selected spectral position; (e) extracted center wavelengths for selected spatial position; (f) first- and second-order statistics of center wavelength across-track profiles; (g) inter/extrapolating curve fitted to center wavelengths for the selected spatial position; and (h)–(k) similar data as in (d)–(g) but for the FWHM parameter.

Center wavelength and FWHM values are approximated for the whole detector by applying a spatial interpolation to the data points already interpolated in spectral dimension. These final layers are compiled into spectral calibration cubes per detector and stored in the database as calibration products, annotated with meta-data describing the parameterization of the calibration module as well as time and date of data

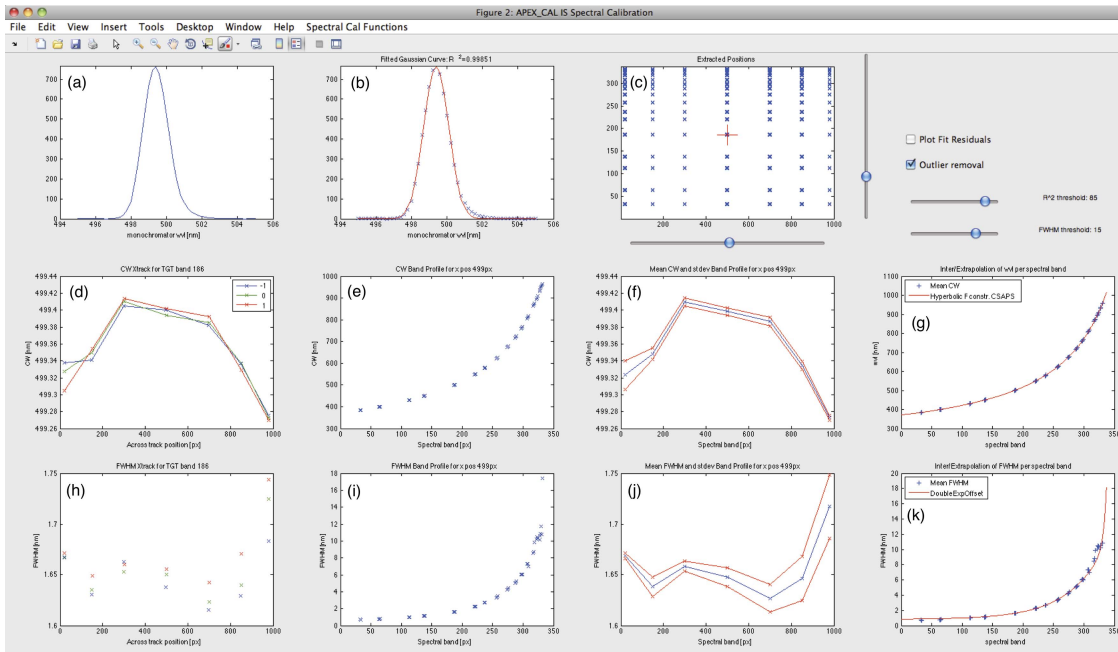


Fig. 11. Interface of the APEX CAL IS spectral calibration module, showing VNIR data.

acquisition and layer generation. At this point, it is readily available to the CAL IS to parameterize operations, such as spectral convolution, required during the radiometric calibration of the system.

VII. DISCUSSION

In the following, we will discuss the selected architectures, namely the database schema and the interface software organized as layers using two programming languages, as these initial choices during system design have an impact on the scalability, flexibility, and speed of the final system, as well as on the implementation effort.

The choice of a flexible data schema, mainly based on the EAV paradigm, over a traditional database schema is a critical one, as it allows a very flexible approach to the handling of meta-data but, on the other hand, database performance may drop considerably as data sizes increase, the latter being one of the main criticisms of the EAV scheme. The flexibility of the system regarding frame-related meta-data could be proven, but careful analysis was needed when considering the modeling of the provenance. Including the provenance information into the EAV concept was initially considered, but the graph structure with processor and specific processing information would not easily fit into the EAV without a massive overhead of logic implemented in the data representation layer. Hence it was decided to add provenance as traditional relational structure, keeping the EAV representation as simple as possible. Data retrieval speed is a matter of the origin of data selection queries. Meta-data are queried extremely fast when starting from the primary datasets (entity-centric operation), while selecting frames based on their meta-data (attribute-centric operation) is more time consuming; this effect is in fact a well known property of EAV databases [31].

For the attribute-centric operation, a careful consideration of the SQL statements is required. Rewriting of queries during implementation to optimize them for the EAV case proved to be essential, sometimes improving the speed by a factor of 10 or more. It was also found that overly complex queries were better split into several consecutive queries, with intermediate results being cached in the EAV database connection and representation layer. An additional concept that proved useful in improving the speed while keeping database statements simple was the sorting or filtering by attributes or attribute values within the application code. The forming of queries as well as the post-query sorting/filtering in the application was implemented as generic as possible within the EAV database connection and representation layer, thus easing and abstracting the data access for the application layer. This strategy resulted in a massive implementation effort in the beginning of the project to code the generic database interface but made consecutive development of calibration specific modules within the application layer extremely efficient.

In hindsight, choosing the EAV paradigm over a traditional relational data model appears to have been the right choice. Data query speeds are time efficient and permit a fluent interaction between user and system, with a scenario dependant 19%–27% of the expected overall data volume already being loaded into the database. This performance is not expected to drop significantly and is in all cases not required to be near real time. Key to the system's performance during attribute-centric queries is the index definition on the values table with indices for `eav_id`, `attribute_id`, `unit_id`, as well as for the actual values. Besides that, most system queries are entity-centric operations with queries being applied to a frame subset and hence not prone to performance deficiencies. The main caveat is the implementation of efficient queries, which may require several iterations, especially while data sizes are

still to increase by several magnitudes. A major advantage of the EAV design proved to be the dynamic generation of new attributes, particularly during the evolution of data processing modules by effortlessly adding new parameters describing the chosen algorithmic configuration. Ultimately, the EAV design allows faster evolutionary development of data processing modules while providing thoroughly acceptable data access speeds for the involved data sizes, provided that queries are optimized for the EAV storage characteristics.

The combination of the programming languages Java and MATLAB proved to be a viable concept. Implementing the EAV database connection and representation layer in Java allowed the reuse of existing classes from the SPECCHIO spectral database project and in turn the addition of the new generic EAV capabilities to it with little overhead involved. The integration of Java within MATLAB was relatively flawless, given the good support of Java by MATLAB and the combination of Java and MATLAB GUI elements with the help of the JControl package in particular, as the latter allowed the development of much more flexible user interfaces than pure MATLAB would have offered. The overhead involved in the loading of frame data represented by serialized Java class instances stored in the database to in-memory instances via database queries and deserialization and the subsequent transfer of data as matrices into MATLAB turned out to be no hindrance as indicated by the loading time results presented, essentially not compromising the required interactive data exploration for databases hosted within the same physical network. Only the loading speed of data from databases via VPN connections may be prohibitive of interactive exploration, mainly caused by the massive amount of data that need transferring for larger frame collections. For these instances, running the application remotely on the server may be the preferred option.

The frame data representation in MATLAB was chosen as the default 64-bit floating point to avoid any conversions during subsequent floating point computations, but a different Java to MATLAB casting, e.g., to 16-bit integer for level-0 and level-1, could be easily added if an increase in the number of frames in memory would be critical. However, all analysis and calibration algorithms written to date have not met such limitations as data subsetting is applied when frame collections grow too big.

VIII. CONCLUSION

The APEX CAL IS is an effective system for the generic storage of imaging spectrometer calibration frame data and associated meta-data including provenance information. It provides a system layer for the database connection and data representation, allowing efficient data access for higher level application programs, such as various calibration and analysis tools whose description is beyond the scope of this paper. The main system components are a MySQL database with an EAV paradigm enabled schema, a system layer implemented in Java, and interactive interfaces written in MATLAB but using Java-based graphical and system layer components. The design was proven to cope very well with the expected amount of data,

and its introduction into the APEX data processing environment resulted in a boost of sensor understanding, calibration to product cycle time, quality control, and repeatability of calibration coefficient estimation.

The rather generic nature of the system suggests that an adaptation to other pushbroom-based systems would be of little effort, requiring only the implementation of appropriate file reading routines for the data insert and the writing of higher level routines for the specific calibration routines required by the target system.

ACKNOWLEDGMENT

The authors would like to acknowledge the financial support by the Swiss Space Office during system design and implementation and the continuing support by EURAMET EMRP ENV-04. The EMRP is jointly funded by the EMRP participating countries within EURAMET and the European Union.

REFERENCES

- [1] S. Ward, *The Earth Observation Handbook—Climate Change Special Edition 2008*. Noordwijk, The Netherlands: ESTEC, 2008.
- [2] *The Second Report on the Adequacy of the Global Observing Systems for Climate in Support of the UNFCCC*, World Meteorological Organization (WMO), Geneva, Switzerland, 2003.
- [3] M. E. Schaepman, S. L. Ustin, A. J. Plaza, T. H. Painter, J. Verrelst, and S. Liang, "Earth system science related imaging spectroscopy—an assessment," *Remote Sens. Environ.*, vol. 113, pp. S123–S137, Sep. 2009.
- [4] *Earth Science and Applications from Space: National Imperatives for the Next Decade and Beyond*, National Research Council, Washington, DC, USA, 2007.
- [5] J. S. Pearlman, P. S. Barry, C. C. Segal, J. Shepanski, D. Beiso, and S. L. Carman, "Hyperion, a space-based imaging spectrometer," *IEEE Trans. Geosci. Remote Sens.*, vol. 41, no. 6, pp. 1160–1173, Jun. 2003.
- [6] M. J. Barnsley, J. J. Settle, M. A. Cutter, and F. Teston, "The PROBA/CHRIS mission: A low-cost smallsat for hyperspectral multi-angle observations of the earth surface and atmosphere," *IEEE Trans. Geosci. Remote Sens.*, vol. 42, no. 7, pp. 1512–1520, Jul. 2004.
- [7] H. Kaufmann, K. Segl, S. Chabrillat, S. Hofer, T. Stuffer, and A. Mueller, "EnMAP a hyperspectral sensor for environmental mapping and analysis," in *Proc. IEEE Int. Conf. Geosci. Remote Sens. Symp.*, Jul.–Aug. 2006, pp. 1617–1619.
- [8] D. Labate, M. Ceccherini, A. Cisbani, V. De Cosmo, C. Galeazzi, L. Giunti, M. Melozzi, S. Pieraccini, and M. Stagi, "The PRISMA payload optomechanical design, a high performance instrument for a new hyperspectral mission," *Acta Astron.*, vol. 65, pp. 1429–1436, Nov.–Dec. 2009.
- [9] T. Stuffer, K. Forster, S. Hofer, M. Leipold, B. Sang, H. Kaufmann, B. Penne, A. Mueller, and C. Chlebek, "Hyperspectral imaging—an advanced instrument concept for the EnMAP mission (environmental mapping and analysis programme)," *Acta Astron.*, vol. 65, pp. 1107–1112, Oct.–Nov. 2009.
- [10] T. Cocks, R. Jenssen, A. Stewart, I. Wilson, and T. Shields, "The HYMAP airborne hyperspectral sensor: The system, calibration and performance," in *Proc. 1st EARSEL Workshop Imag. Spectroscopy*, 1998, pp. 1–6.
- [11] M. E. Schaepman, M. Jehle, A. Hueni, K. Meuleman, and K. I. Itten, "The 4th generation imaging spectrometer APEX and its application in earth observation," *IEEE Trans. Geosci. Remote Sens.*, 2013, to be published.
- [12] R. O. Green, M. L. Eastwood, C. M. Sarture, T. G. Chrien, M. Aronsson, B. J. Chippendale, J. A. Faust, B. E. Pavri, C. J. Chovit, M. Solis, M. R. Olah, and O. Williams, "Imaging spectroscopy and the airborne visible/infrared imaging spectrometer (AVIRIS)," *Remote Sens. Environ.*, vol. 65, pp. 227–248, Jul. 1998.
- [13] F. Lehmann, D. Oertel, R. Richter, P. Strobl, H. Rothfuß, A. Mueller, P. Hausknecht, S. Tischler, R. Mueller, D. Beran, J. Fries, R. Boehl, and P. Obermeier, "The digital airborne imaging spectrometer DAIS 7915," in *Proc. ISSIR Hyperspectral Appl. New Sensor*, 1995.

- [14] H. Kostkowski, *Reliable Spectroradiometry*. La Plata, MA, USA: Spectroradiometry Consulting, 1997.
- [15] E. J. Milton, M. E. Schaepman, K. Anderson, M. Kneubühler, and N. Fox, "Progress in field spectroscopy," *Remote Sens. Environ.*, vol. 113, pp. 92–109, Sep. 2009.
- [16] P. Cox and D. Stephenson, "A changing climate for prediction," *Science*, vol. 317, no. 5835, pp. 207–208, 2007.
- [17] N. Fox, J. Aiken, J. J. Barnett, X. Briottet, R. Carvell, C. Frohlich, S. B. Groom, O. Hagolle, J. D. Haigh, H. H. Kieffer, J. Lean, D. B. Pollock, T. Quinn, M. C. W. Sandford, M. Schaepman, K. P. Shine, W. K. Schmutz, P. M. Teillet, K. J. Thome, M. M. Verstraete, and E. Zalewski, "Traceable radiometry underpinning terrestrial- and helio-studies (TRUTHS)," *Adv. Space Res.*, vol. 32, no. 11, pp. 2253–2261, 2003.
- [18] R. L. Ackoff, "From data to wisdom," *J. Appl. Syst. Anal.*, vol. 16, no. 1, pp. 3–9, 1989.
- [19] S. Kempler, C. Lynnes, B. Vollmer, G. Alcott, and S. Berrick, "Evolution of information management at the GSFC earth sciences (GES) data and information services center (DISC): 2006–2007," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 1, pp. 21–28, Jan. 2009.
- [20] L. Floridi, "On defining library and information science as applied philosophy of information," *Soc. Epistemol.*, vol. 16, no. 1, pp. 37–49, 2002.
- [21] L. Floridi, "Understanding epistemic relevance," *Erkenntnis*, vol. 69, no. 1, pp. 69–92, 2008.
- [22] J. Rowley, "The wisdom hierarchy: Representations of the DIKW hierarchy," *J. Inf. Sci.*, vol. 33, no. 2, pp. 163–180, 2007.
- [23] C. W. Choo, "The knowing organization: How organizations use information to construct meaning, create knowledge and make decisions," *Int. J. Inf. Manage.*, vol. 16, no. 5, pp. 329–340, 1996.
- [24] D. Chaffey and S. Wood, *Business Information Management: Improving Performance Using Information Systems*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2005.
- [25] A. Hueni, J. Biesemans, K. Meuleman, F. Dell'endice, D. Schläpfer, S. Adriaensen, S. Kempenaers, D. Odermatt, M. Kneubuehler, J. Nieke, and K. Itten, "Structure, components and interfaces of the airborne prism experiment (APEX) processing and archiving facility," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 1, pp. 29–43, Jan. 2009.
- [26] P. Gege, J. Fries, P. Haschberger, P. Schötz, H. Schwarzer, P. Strobl, B. Suhr, G. Ulbrich, and W. J. Vreeling, "Calibration facility for airborne imaging spectrometers," *ISPRS J. Photogram. Remote Sens.*, vol. 64, pp. 387–397, 2009.
- [27] M. Ecker, "Automatisierung einer optoelektronischen kalibriereinrichtung," M.S. thesis, Dept. Appl. Sci., Landshut Univ., Landshut, Germany, 2007.
- [28] T. D. Wason and D. Wiley, "Structured metadata spaces," *J. Internet Catalog.*, vol. 3, nos. 2–3, pp. 263–277, 2000.
- [29] D. Bernholdt, S. Bharathi, D. Brown, K. Chanchio, M. Chen, A. Chervenak, L. Cinquini, B. Drach, I. Foster, P. Fox, J. Garcia, C. Kesselman, R. Markel, V. Nefedova, L. Pouchard, A. Shoshani, A. Sim, G. Strand, and D. Williams, "The earth system grid: Supporting the next generation of climate modeling research," *Proc. IEEE*, vol. 93, no. 3, pp. 485–495, Mar. 2005.
- [30] P. Nadkarni, L. Marenco, R. Chen, E. Skoufos, G. Shepherd, and P. Miller, "Organization of heterogeneous scientific data using the EAV/CR representation," *J. Amer. Med. Informat. Assoc.*, vol. 6, no. 6, pp. 478–493, 1999.
- [31] V. Dinu and P. Nadkarni, "Guidelines for the effective use of entity-attribute-value modeling for biomedical databases," *Int. J. Med. Informat.*, vol. 76, nos. 11–12, pp. 769–779, 2007.
- [32] H. Arndt, M. Bundschuh, and A. Naegele, "Toward a next-generation matrix library for java," in *Proc. 33rd Annu. IEEE Int. Computer Softw. Appl. Conf.*, Jul. 2009, pp. 460–467.
- [33] A. Hueni, J. Nieke, J. Schopfer, M. Kneubühler, and K. Itten, "The spectral database SPECCHIO for improved long term usability and data sharing," *Comput. Geosci.*, vol. 35, no. 3, pp. 557–565, 2009.
- [34] M. Lidiether, "sigTOOL: A MATLAB-based environment for sharing laboratory-developed software to analyze biological signals," *J. Neurosci. Methods*, vol. 178, no. 1, pp. 188–196, 2009.



Andreas Hueni (M'12) received the B.Sc. degree in computer science from the University of Applied Science Brugg-Windisch, Brugg-Windisch, Switzerland, in 1997, the PG Diploma degree in geographic information systems, and the M.Phil.(Sc) degree in earth science from Massey University, Palmerston North, New Zealand, in 2005 and 2006 respectively, and the Ph.D. degree in geography from the University of Zurich, Zurich, Switzerland, in 2011.

He is currently a Post-Doctoral Researcher with the Remote Sensing Laboratories, University of Zurich, where he is involved in research on APEX sensor and data calibration. He was involved in research for the APEX project in 2007. His current research interests include calibration of spectrometers and design of combined databases and software systems such as spectral database SPECCHIO.



Karim Lenhard received the Diploma degree in physics from the University of Bonn, Bonn, Germany, in 2008, where he is currently pursuing the Ph.D. degree with the German Aerospace Center (DLR).

He is currently a Researcher with DLR where he is engaged in operating and improving the APEX calibration home base, an optical laboratory for calibration, and characterization of airborne hyperspectral imagers. His current research interests include spectral and radiometric calibration of hyperspectral

instruments.



spectrometers.

Andreas Baumgartner received the B.Eng. degree in mechatronics specialized in optical engineering and the M.Eng. degree in electrical engineering and information technology from the University of Applied Sciences Deggendorf, Deggendorf, Germany, in 2008 and 2010, respectively.

He is currently with the German Aerospace Center where he is engaged in calibration home base of APEX. His current research interests include improving setup, methods, and software for spectral, geometric, and radiometric calibration of imaging



Michael E. Schaepman (M'05–SM'07) received the M.Sc. and Ph.D. degrees in geography from the University of Zürich (UZH), Zürich, Switzerland, in 1993 and 1998, respectively.

He is currently the Head of the Remote Sensing Laboratories, Department of Geography, UZH, where he joined as a Full Chair of remote sensing in 2009. In 1999, he was a Post-Doctoral Researcher with the Optical Sciences Center, The University of Arizona, Tucson, USA. In 2000, he was a Project Manager with the European Space Agency Airborne Prism Experiment spectrometer. In 2003, he was a Full Chair of geoinformation science and remote sensing with Wageningen University, Wageningen, The Netherlands. His current research interests include computational Earth sciences using remote sensing and physical models, particularly land-atmosphere interface using imaging spectroscopy.